Coupling World-class Computer Science with World-class Applications: the Advanced Computing Laboratory

Rod Oldehoeft - Deputy Director

November 22nd, 1999

The ACL - an integrated high-performance computing simulation effort:

Qualitatively Distinct Applications

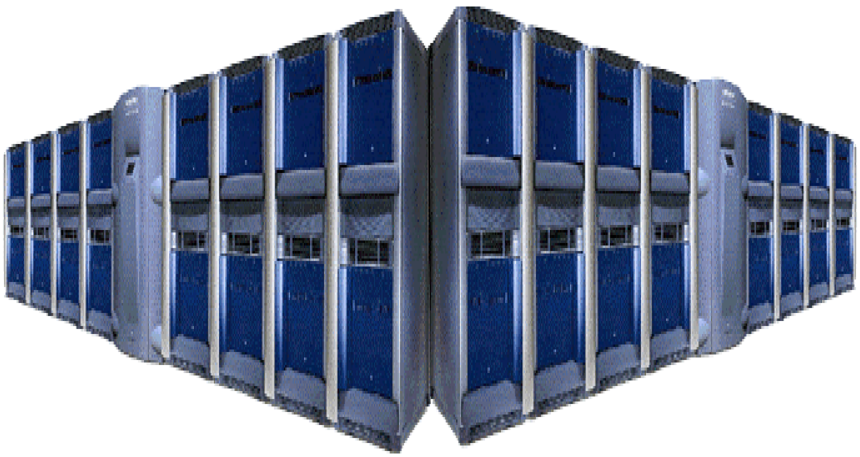
High-Performance Scalable Platforms

High-Performance Scalable Software

Must have all three working in concert to progress towards strategic simulation objectives

The ACL Nirvana Machine: the Largest Dedicated, Open Computing Platform in the Nation

- 2,048 processor SGI/Cray Origin 2000 system accepted on June 24th
- Peak Speed of 1 trillion floating point operations per second
- 500 billion Bytes of memory
- 7 trillion Bytes of disk storage
- 100 trillion Bytes of tape storage
  - several libraries of congress

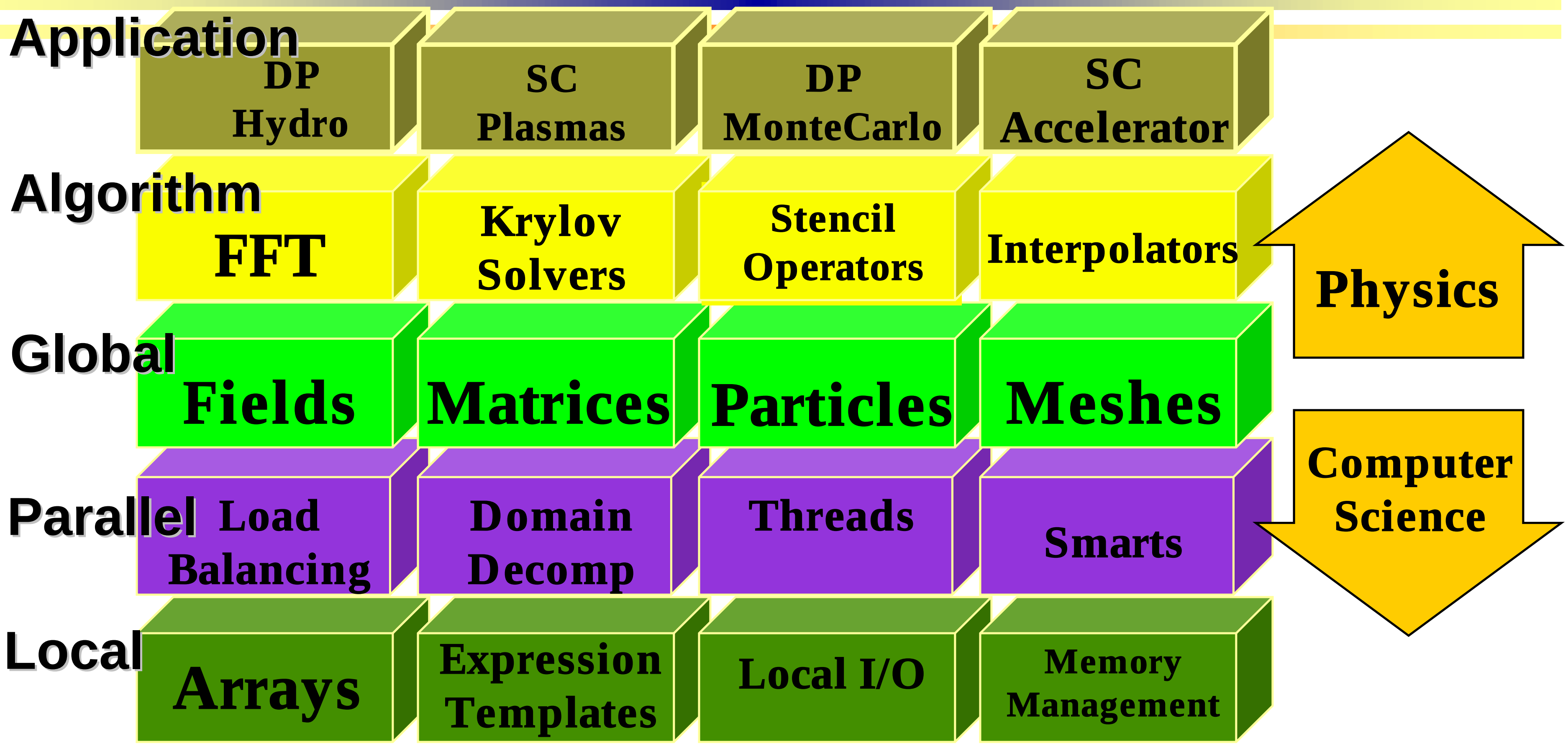


- 16 Infinite Reality Graphics Engines integrated into the system to provide the largest unclassified scientific visualization capability in the world
  - real-time interaction with 1 Billion cell data-sets
- The ACL Nirvana machine is focused upon Capability Computing
  - Over the past two months, over 40% of the jobs on the ACL Nirvana machine have run on at least 512 processors; at least 20% of the jobs have used 1024 processors
  - Several users have run applications using the full 2,048 processors available

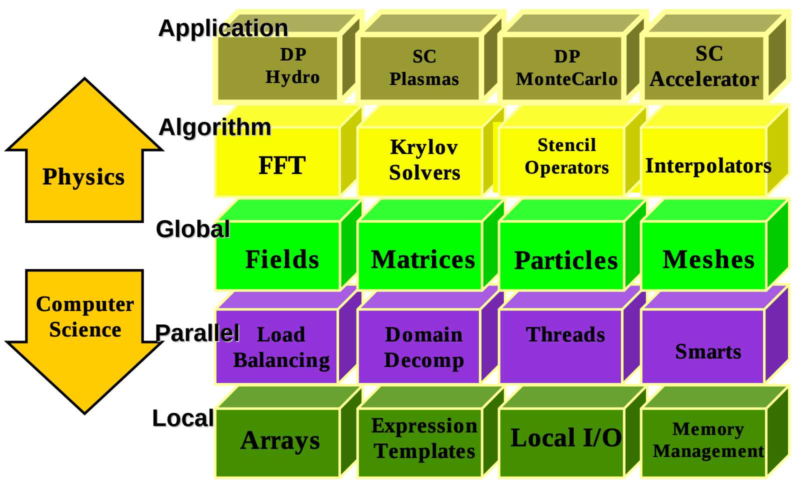
Our Computer Science Efforts

- ***Parallel object-oriented techniques***
- ***Scalable run-time systems***
- ***Extreme Linux computing***
- Component architectures
- Advanced systems science
- Distributed computing
- High-performance networks
- Computational mathematics
- Scalable visualization

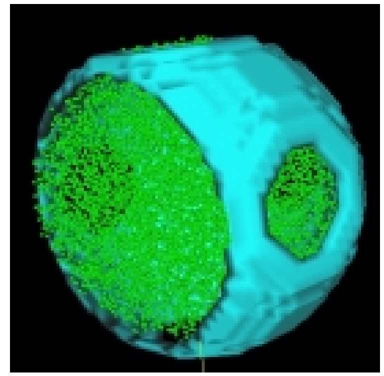
The POOMA FrameWork



The POOMA Framework



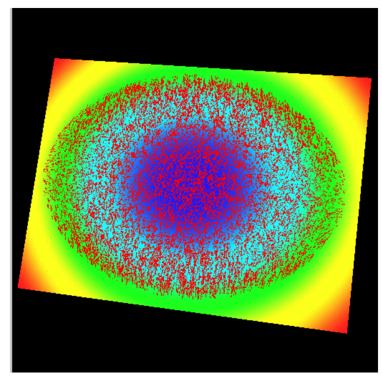
Reused particle classes from the SC NTP Grand Challenge to build the first parallel ASCI simulation to run across all three ASCI computers



Neutronics simulation of multi-material shell

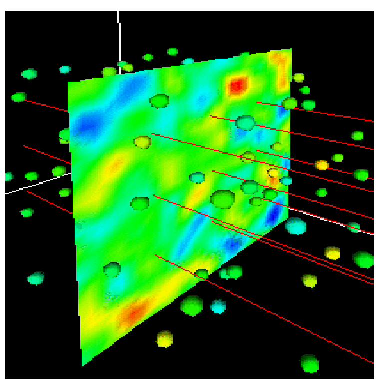
Developed on a SunOS workstation and ported with no changes to three parallel ASCI platforms

Parallel particle classes were optimized and deployed back into a new SC Grand Challenge - Advanced Accelerators



Propagation of an intense beam in a 3D periodic transport system of magnetic quadrupoles used to study the formation of halos in mis-matched charged particle beams.

Geometry optimizations made to the accelerator simulation reused to enable Numerical Tokamak Turbulence Project SC Grand Challenge simulations

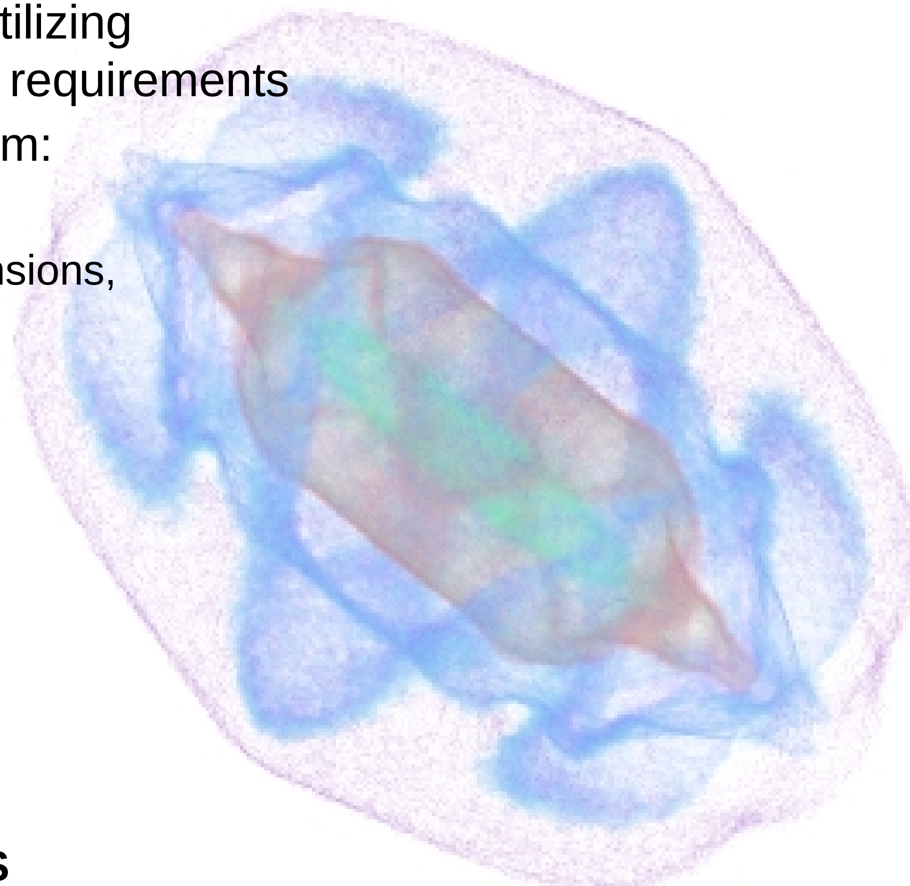


3D Gyrokinetic simulation of an ion-temperature gradient driven instability.

POOMA Framework grew from particles classes used to build parallel plasma simulations in support of the SC Numerical Tokamak Project (NTP) Grand Challenge

Computational Accelerator Physics Grand Challenge

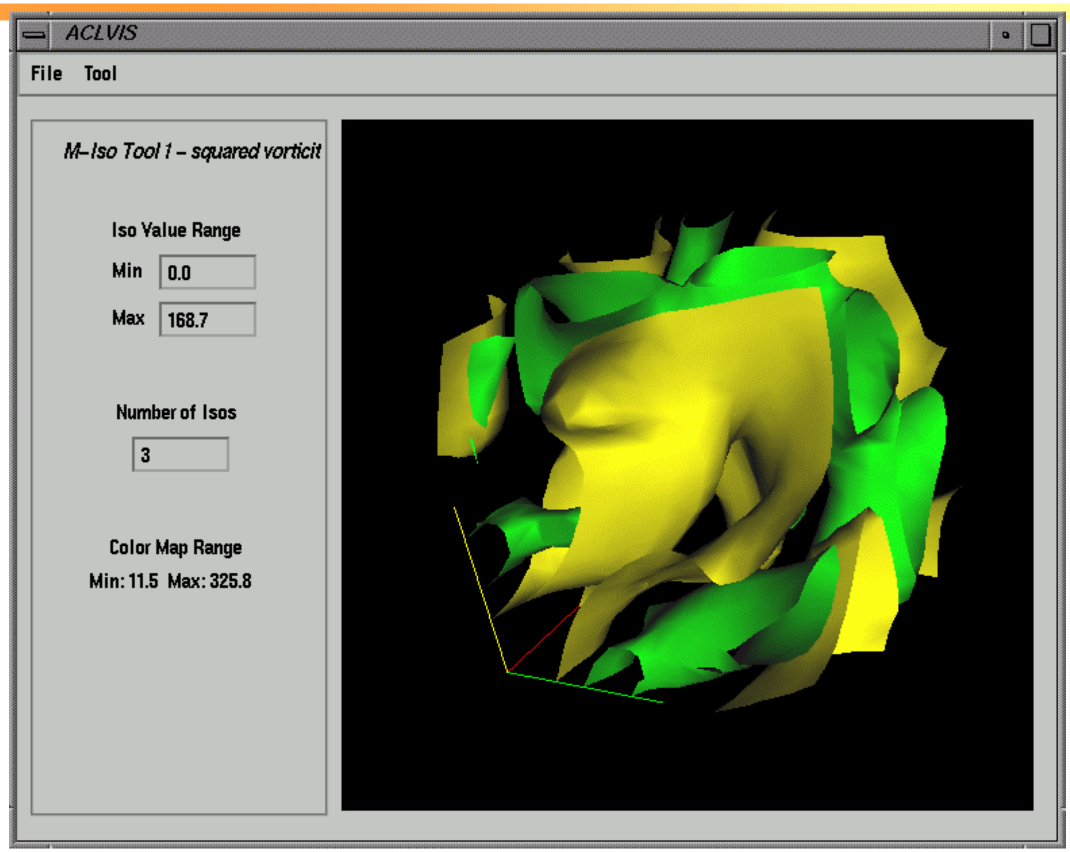
- Accelerator Grand Challenge project utilizing POOMA for parallel particle simulation requirements
- Significant involvement of POOMA team:
  - Port of **IMPACT** to **POOMA**
  - Many **POOMA** modifications and extensions, such as:
    - optimized FFTs
    - modifications for tree solver
  - Performance optimization
- IMPACT ported to PC cluster
- Support from Visualization team
  - Parallel volume rendering
  - Parallel runtime viz
- IMPACT demo as SC99
  - communication among parts via **PAWS**



Rapid Application Development with POOMA

- Fluid Simulation
- 3D Pseudo-Spectral
- Navier-Stokes Equations
- Parallel I/O
- Run-Time Visualization

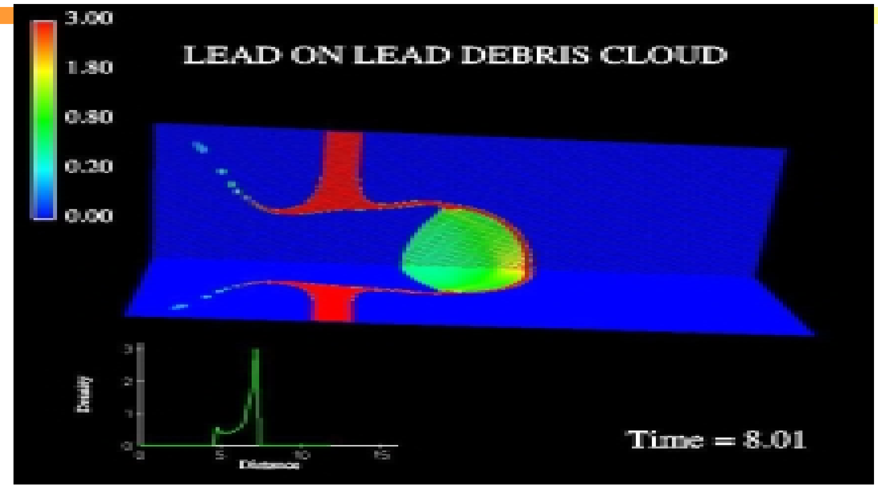
- *One Post-Doc with no parallel experience wrote this application in 5 weeks with POOMA*

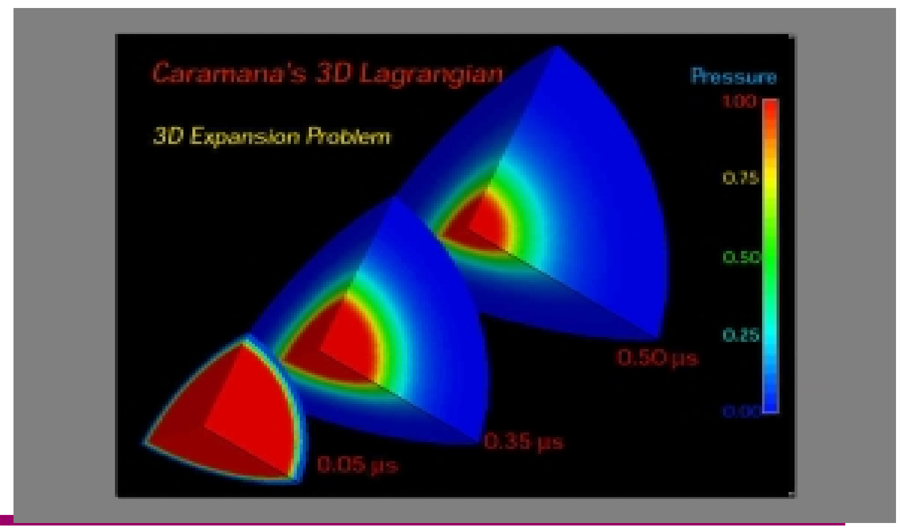


Navier-Stokes simulation iso-surface of vorticity

ASCI: Multi-Material Hydrodynamics

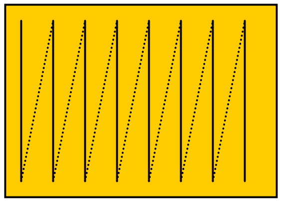
- ASCI Blanca project utilizing POOMA for all parallel simulation requirements
- Blanca is Los Alamos' lead effort in simulating weapons safety and performance
- Rapid application development:
  - Blanca was originally designed with Eulerian capabilities
  - Utilizing the generic programming features of POOMA, a Lagrangian capability was added to Blanca in two weeks
  - The first test run resulted in a successful, scalable parallel simulation





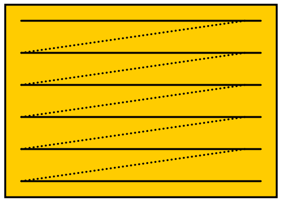
Many Possible Array Implementations: Single Interface - Generic Programming

- Fortran Arrays
  - Dense storage
  - First index varies fastest



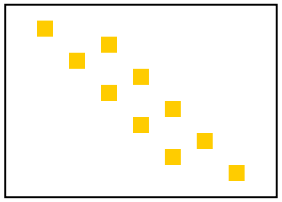
Fast random access
Full storage
Not aliased

- C Arrays
  - Dense storage
  - Last index varies fastest



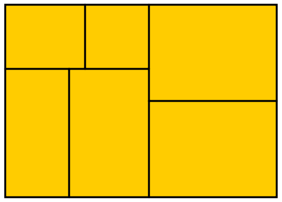
Fast random access
Full storage
Maybe aliased

- Sparse Arrays
  - Small fraction nonzero
  - Row compressed, list, etc



Sequential access
Reduced storage

- Multipatch Arrays
  - Store sections separately
  - Each section is an Array



Slow random access
Fast random w/in patch
Distributed storage

POOMA II: Incremental Adoption

- POOMA I controlled the data
- You could put Fortran in a POOMA code, but not the other way around.
- With an appropriate Engine, a POOMA II Array can attach to existing data.
- Add POOMA modules to an existing code.
- Easier adoption
- Compartmentalization of advanced techniques.

Old style: monolithic

POOMA I code

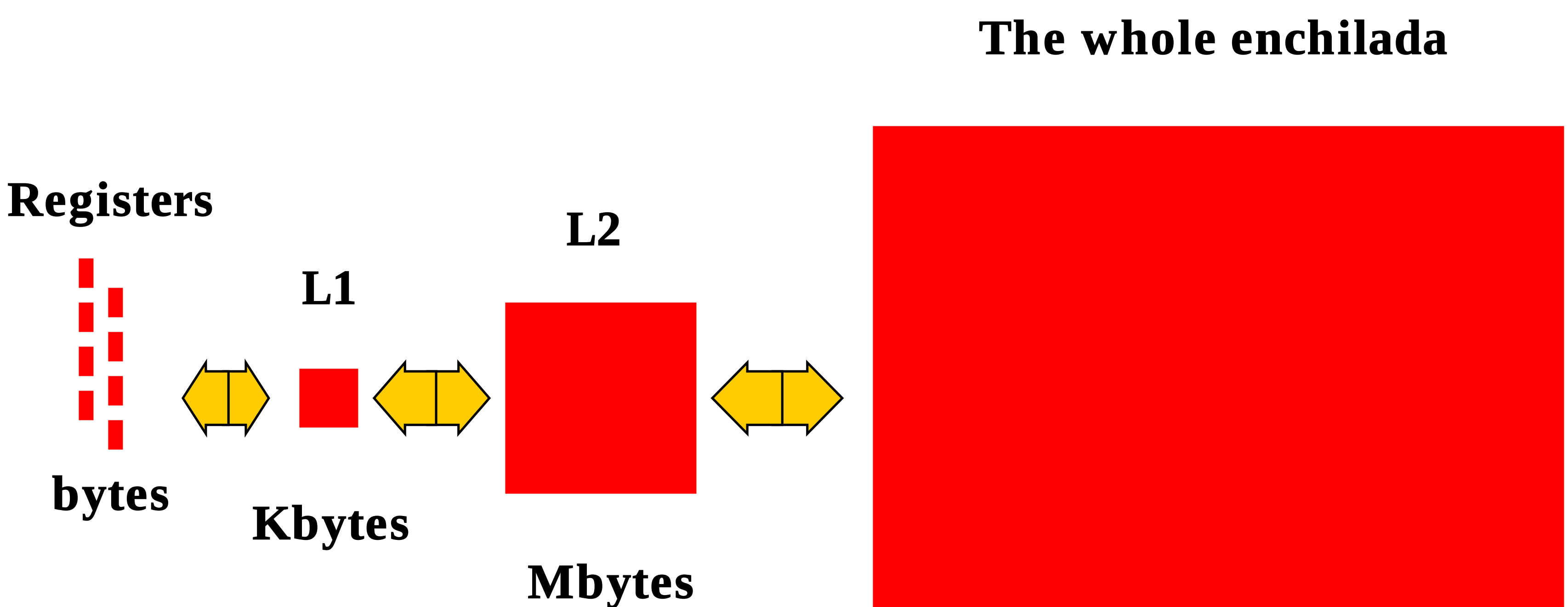
Fortran module

New style: adaptable

Fortran code

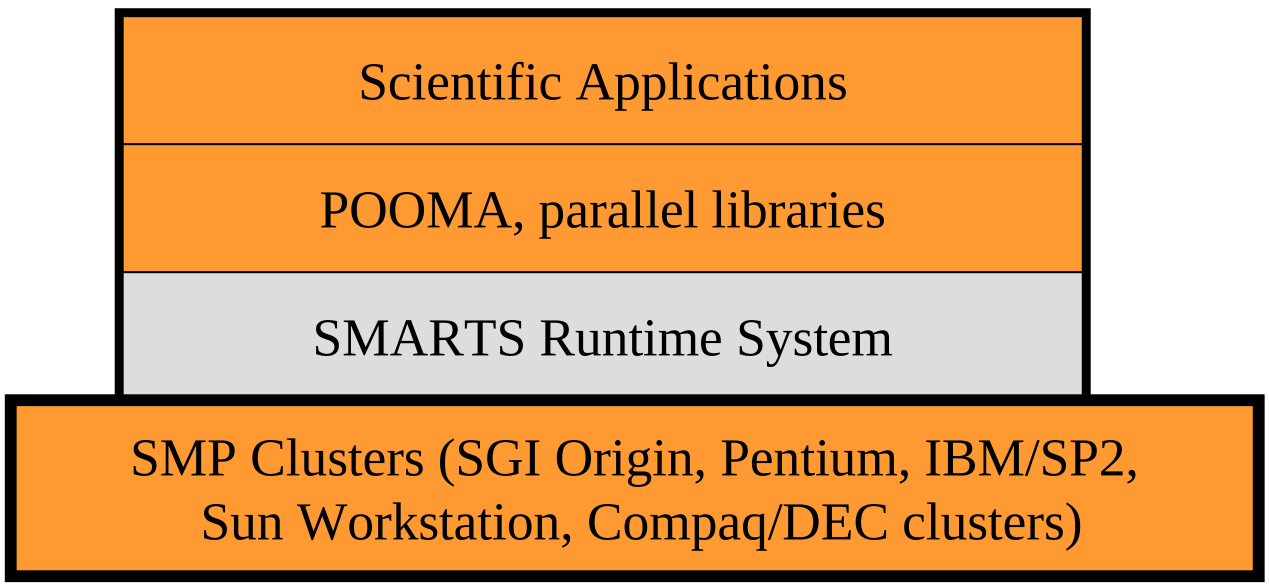
POOMA II module

The Memory Wall: Memory Hierarchy



- One->tens->hundreds of cycles
- Wulf's memory wall is here!!
- Memory is a large part of machine costs: not a scalable commodity solution

Middleware for Scientific Applications



SMARTS:

scalable multi-threaded asynchronous run-time system

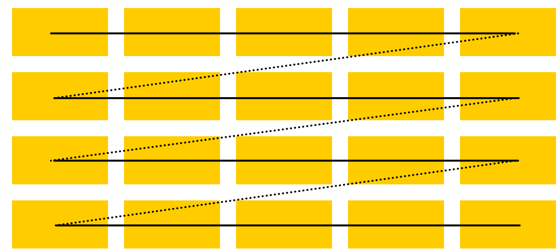
- Runtime system for scientific applications
  - supports high-performance library/framework development on clustered SMP architectures
- Asynchronous Dataflow Model
  - permits a high degree of parallelism.
- Designed for Clusters of SMP’s
  - deep memory hierarchies
  - registers, multiple levels of cache, local memory, and remote off-box memory
  - **SGI/Pentium Clusters**

Nano-tasking Evaluation with POOMA and SMARTS

- **Large array operations, although natural for scientific expression, are the worst possible thing for cache based machines.**
  - **Process whole arrays at once instead of in pieces.**
- **By evaluating out of order, we can preserve cache and simplicity of expression.**
- **Iteration-space tiling, figured out at run-time.**

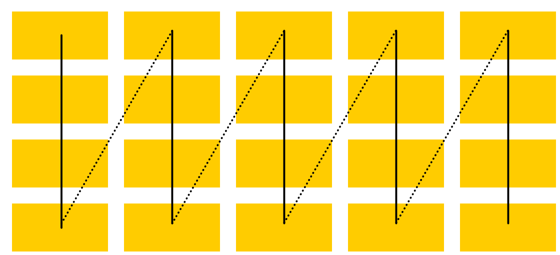
Conventional Array Operations

```
A=B+C*D;
B=X+Y*A;
C=D+X*B;
D=Y+X*C;
```

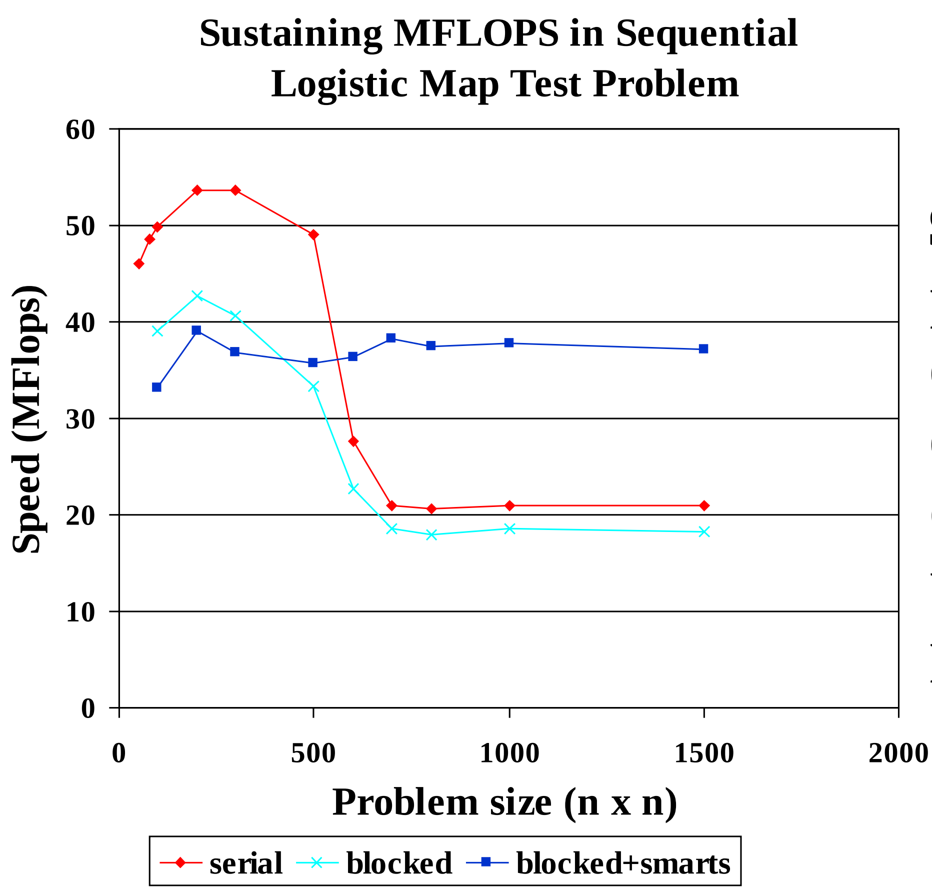


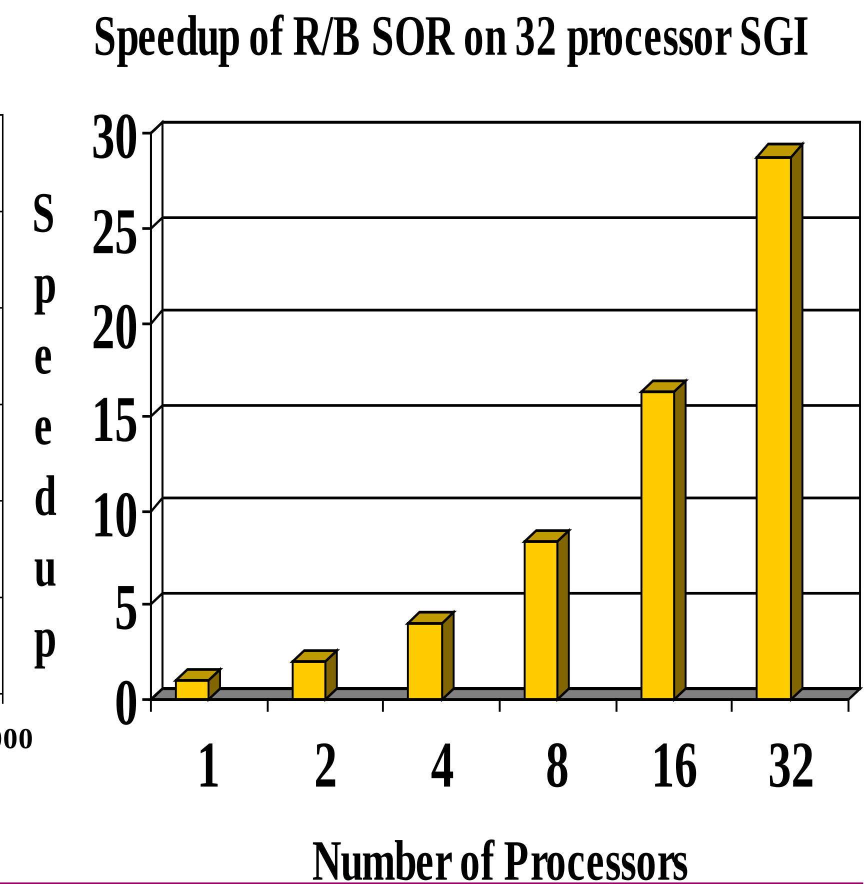
With SMARTS Dataflow

```
A=B+C*D;
B=X+Y*A;
C=D+X*B;
D=Y+X*C;
```



Preliminary Performance Results





Little Blue Penguin Cluster

- 128 Pentium II 333Mhz CPUs arranged as 64 SMP nodes
- Myrinet high-speed interconnect
- 1/2 Terabyte RAID disk farm
- 32 Gigabytes RAM
- 320 Gigabyte local scratch disk

The ACL and collaborators have ported all Office of Science supported software to Linux clusters:
Tulip, PAWS, SILOON, POOMA, TAU, PDT, Ductape, …

The Penguins currently run several Office of Science Applications:
Accelerators, QCD, Oceans, ...

Extreme Linux Leadership

- Strong Office of Science collaboration among Argonne, Berkeley, and Los Alamos.
- SC'99 tutorial
- Ongoing software development projects for scalable Linux cluster application support
- Exploring the viability of Linux cluster platforms for future TeraScale computing
- Strong collaborations with DP, NSA, and NSF
- DOE currently has leadership in this area!!

Observations

- The complexity of current and future computing platforms precludes conventional software approaches
- Applications are simulating a greater complexity of phenomena than ever before
- Successful strategic simulations require an effective blend of computer and computational sciences